Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 3 Issue 7, July - 2016

A Secure Logbook For Bluetooth Data Transfer On Mobile Devices

Oyinloye Oghenerukevwe. E
Computer Science Department
Ekiti State University,
Ado – Ekiti, Nigeria
rukkivie@yahoo.com

Ojedayo Benson. O
Computer Engineering Department
Ekiti State University
Ado – Ekiti, Nigeria.
bentheta@yahoo.com

***Abstract*—Security is one of the major issues in Bluetooth communication. As Bluetooth works in unlicensed frequency band, it is more vulnerable to many security attacks. Bluetooth's inability to keep a permanent log detail of transfers; this weakness has made users unaware about intruders' unauthorised access to files/documents by cleaning the conventional Bluetooth log. This research paper presents an offline stand-alone application called Bluelog aimed at creating an efficient and secured working environment for storing or retrieving data sent via Bluetooth.**

Keywords—Bluetooth, wireless communication, Bluetooth Low Energy, Server side, Client side, Database, JBPeer, Bluelog

I. INTRODUCTION

Wireless communications is, by any measure, the fastest growing segment of the communications industry. As such, it has captured the attention of the media and the imagination of the public [1]. Mobile phones have experienced remarkable development over the last decade, and this growth continues unabated worldwide, with more than a billion worldwide cell phone users projected in the near future. Indeed, mobile phones have become a critical business tool and part of everyday life in most developed countries, and are rapidly superseding old wire line systems in many developing countries. The explosive growth of wireless systems coupled with the proliferation of laptop and palmtop computers indicate a brighter future for wireless networks, both as stand-alone systems and as part of the larger networking infrastructure.

Bluetooth wireless technology is a short-range communications technology intended to replace cables connecting portable and or fixed devices while maintaining high levels of security. The key features of Bluetooth technology are robustness, low power, and low cost. There are many different Bluetooth applications which have been commercially developed. Bluetooth can be used to connect mobile phones to offices, landline telephone sets and employees can make calls even when not at their desk from their landline number with the help of their Bluetooth enabled phone. Bluetooth in all electronic devices, development into such applications is cost efficient and has become mandatory for achieving short range wireless connectivity [2].

II. HISTORY OF BLUETOOTH TECHNOLOGY

Bluetooth (named after Harald Bluetooth, a tenth century king who unified Denmark and Norway) is an open standard for short-range ad hoc wireless voice and data networks, operating in the unlicensed ISM 2.4 GHz frequency band.

Bluetooth was originally conceived by Ericsson in 1994**.** In 1998, Ericsson, Nokia, IBM, Intel, and Toshiba formed a special interest group (SIG) to expand the concept and to develop a standard under IEEE 802.15; of which Currently there are over 2000 companies participating in the Bluetooth SIG, and many are developing Bluetooth products. The Bluetooth SIG considers three application scenarios.

The first is wire replacement, to connect a PC or laptop to its peripherals.

The second is the ad hoc networking of several different users at short ranges in a small area, forming a "piconet," similar to but smaller than an IEEE 802.1 1 cell.

The third is to use Bluetooth as an access point to wide-area voice and data services provided by a wired network or cellular system.

The last two application scenarios are in direct competition with the intended use of IEEE 802.1 1[3].

Bluetooth transmits at 1mW (0 dBm), using hybrid direct sequence spread spectrum and frequency-hopping spread spectrum technologies. It can accommodate up to three voice channels or seven data channels per piconet, and a data speed of 721 Kbps per piconet. Its expected system range is around 10 meters. It can support up to eight devices per piconet, and 10 piconets in a given coverage area. It can provide some security at the link layer, and requires 2.7 V as a power source. Finally, a Bluetooth device consumes 30μA in sleep mode, 60 μA in hold mode, 300 μA in the standby mode, and 8-30 mA while transmitting [3].

III CLASSIC BLUETOOTH AND Bluetooth Low Energy (BLE)

Although BLE and classic Bluetooth use the same spectrum range (the 2.400 GHz-2.4835 GHz ISM band), they use different sets of channels. BLE uses only 40 (2MHz) channels, whereas classic Bluetooth uses 79 (1MHz). In order to mitigate interference

caused by crowded bands, both technologies use a technique called frequency hopping, albeit with different details. For devices to be discovered there are three RF channels available for advertisements that are allocated in different parts of the spectrum to avoid interference from Wi-Fi [4]

Another difference between classic Bluetooth and BLE are the data transfer rates. Classic Bluetooth, in its original release, could transfer data at 1Mbit/s. It had gone up to 3Mbit/s with the release of Bluetooth 2.0 Enhanced Data Rate version, and can reach even faster speeds with the High-Speed version of Bluetooth 3.02. With Bluetooth 4.0, however, a fast data transfer rate was dropped in favors of lower power consumption. To achieve this, BLE sleeps for longer periods of time, send bursts of data less frequently, and it does not maintain connections, like classic Bluetooth, but can quickly re-establish links when communication is needed again. Services that stream music, for example, will still be used by classic Bluetooth instead of the new low energy version [5].

Additionally, because of the low power consumption, BLE has a smaller time window to send data in. With classic Bluetooth having 100ms, and BLE less than 3ms, justifying the fact that BLE was designed to send less data than traditional Bluetooth (Laird technologies, 2014). To ensure that the data transferred in this time is not redundant, the Generic Attribute Profile (GATT) that is specific to Bluetooth v4.0 is used as a specification for data transmissions over BLE links. The Bluetooth SIG defines profiles for BLE devices, which describes how a device works in a certain application [5]

IV. EXISTING SYSTEM

Base on the research done, some Bluetooth applications have already been in existence, these applications are targeted at solving and achieving different task, below are examples on existing applications related to the use of Bluetooth.

V. BLUETOOTH MESSENGER

The researchers developed an application which connects two or more android smart phones, the enables them to send one-to-one messages via Bluetooth. Bluetooth connection between a server and an android smart phone whereby the user can update and synchronize his/her chat records with the server from time to time. Data structures were used in storing and updating the data (messages) against respective usernames. The application was designed using python programming language.

The application significantly uses Bluetooth to achieve this goal. The application first checks if the phone's Bluetooth are ON and if it's not the application would request the user to give permission to turn it ON. After the phone's Bluetooth is checked, if it is the first time the application is being run it will ask the user to enter its name. This name which is entered by the user will be sent during the Introduction section of messaging where the users are introduced to one another by the name they take up. A file 'udetails.txt' is created recording the user's input. This data is then fetched and sent to every phone with which the user communicates. This feature gives a better identification of the user since while scanning for nearby Bluetooth devices it is often very difficult to identify the user.

The application has essentially three types of terminals: The communicating 'server phone' terminal, the communicating 'client phone' terminal and he 'storage server' terminal. The server and client terminal form the two phones involved in messaging whereas storage sever terminal is the Android based storage server.

The benefit of the system is that two to more phones can be connected together for chatting. Installation is a hassle-free task and the user interface is user friendly and does not require specialized training or skills to operate it.

The limitations of the system are that It requires internet connection for one time to install the application and also enhancing security by encryption [8]

VI. MOBILE P2P FILE SHARING ENVIRONMENT OVER BLUETOOTH

The authors developed a system that is unique in that all other mobile applications that offer file sharing services with the involvement of a file server. This involved heavy download and uploads to a file server and because this is being carried out over the GSM networks, the users are charged [10]. Figure 6.1 describes the JBPeer system as a system which implements two optional API's (Application Program Interface) that are not part of the core J2ME platform. Small, resource limited devices implement the Connected Limited Device Configuration (CLDC). The CLDC is a specialist set of API's designed to extend the functionality provided by the basic J2ME platform. Some APIs are left as optional by Sun Microsystems. The two optional APIs that a device must implement if it is to run the JBPeer MIDlet is JSR-82 and JSR-75, which are for Bluetooth and File System access respectively.

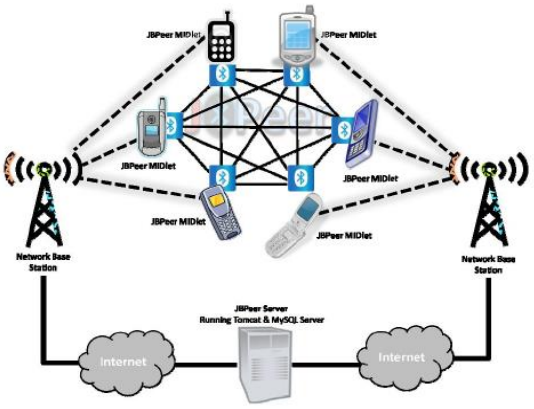

Figure 1 Architecture of JBpeer System [10]

The architecture of the system shown in Figure 1 follows that of most desktop based peer to peer systems in that each node is equal and performs both client and server functionality. However the existence of the server creates a slightly different architecture from the standard P2P architecture in which no central server exists.

The purpose of a central server in this system is to allow the users to let the server know what resources they want to share and allow other users to search the server for device file lists. Without a central server, each device is responsible for searching the locally available Bluetooth devices and then attempting to query each device for its file list which would create a huge amount of communication overhead. This would cripple the application, making it too slow to use and perhaps even run. The use of a central server will remove the need to implement a P2P protocol on a resource limited platform.

During the account creation process the user is asked to enter the Bluetooth M\AC address of their device, which is shown on the JBPeer MIDlet login screen. This MAC address is used along with the username as a hash key to hash out the users password.

This information is then stored to the database. In adopting this technique, the user can only authenticate onto the JBPeer system if they do so from the device they registered with. This makes it almost impossible for malicious users to gain access to another person's account. Once the user has created them an account they will then be able to log into the system through the JBPeer MIDlet. This authentication will require the transmission of a small amount of data over the GSM network. If authentication is successful then the MIDlet immediately ask the user to set a directory, in their mobile devices file structure, that they want to share. If the user has previously logged into the system before then the location they set previous is remembered. With the shared directory set, the MIDlet then starts the Bluetooth server and creates a list of all the filenames that are located in the shared folder and sends that list to the server. This communication requires the transmission of data over the GSM network. At the server end, each filename that is received is added to a file table and the Bluetooth M\AC address of the device that sent it is also added. The purpose of adding the M\AC address to each file is so that when a query is performed, only the filenames of the device that is discovered are returned.

The user is then able to perform a Search. In doing so the MIDlet attempts to locate any device within its range that is currently running the JBPeer MIDlet as well. A list of discovered devices is then presented to the user. The list itself is simply the Bluetooth M\AC addresses of the devices found. The reason for making the list display M\AC address rather than the 'Friendly name' of the device is to ensure anonymity which is provided by most desktop P2P systems. The user can then select a device whose file list they want to see. This action creates a query to the Central server which requires the transmission of data over the GSM network.

The limitation of the system is the long distance. This can be seen as a limitation or in the case of Mobile Social Software (MoSoSo) applications; this could be exploited to help bring strangers together through their common interest in files.

VII. BLUETOOTH ORDERING APPLICATION

The researcher developed an application which is wirelessly deployed on the users mobile whenever a person enters the specified area such as a restaurant, the application enables the user to perform activities like viewing the menu with prices, and to place order or to complain as well as make suggestions for managers. The administrator of the application is entitled to see the list of orders, to change the prices of food and to maintain the services provided by the management. The customer can only make payment in cash.

The application functions in actual working environment. First level of the system shows that Bluetooth uses radio signals to communicate with other Bluetooth device. In the second level the operating system checks and installs Bluetooth hardware driver and after that application detects Bluetooth stack in underlying system. Operation of the application performs when client is searched in the range of the server. Plus the client should have a Bluetooth-enabled mobile, and after the client confirms the request of the server, client can make the benefit of the services that are provided by the restaurant server to the client which automatically installs the client application on the client's mobile [7]. The application contains the restaurant menu, details of the services offered, prices and types of the

pizzas. User can then place orders, can see the rate list and after placing the order he also receives the billing information.

All the data that is coming from the client side is saved in the data base that is maintained on the server side. The application was achieved using the Nokia mobile classic 6700 which the activities really worked properly on [7].

The limitation of the application occurs at the client's side when getting database records, From pizza restaurant server only little rate list and menu are displayed because only few libraries are available in J2ME for small devices [7].

VIII. A Secure Logbook for Bluetooth Data Transfer on Mobile Devices

i. SYSTEM DEVELOPMENT

The program is developed for a Bluetooth logbook to store details information via Bluetooth connection between different devices. The system is implemented in Android SDK. Android is a software stack for mobile devices that includes an operating system, middleware and key Android SDK provides the tools and APIs begin developing applications on the Android platform using the Java programming language. It has the features of SQLite for structured data storage, Application framework enabling reuse and replacement of components, optimized graphics powered by custom 2D graphics library;3D graphics based on the OpenGL ES 1.0 specification and supports GPS, Compasses, Accelerated 3D graphics,EDGE,3G,WiFi,SMS messaging, MMS, Bluetooth, Video/still digital cameras and touch screen. Includes set of development tools like Debugger, Libraries, An emulator, Documentation, Sample code and Tutorials.

The designed Secured Bluetooth transfer logbook called Bluelog is designed based on two-tier architecture; which is a client-server model as shown in figure 2 while figure 4 shows the Sever-Client device Bluetooth connection's Sequence Diagram
The flowchart of system log activity is thus shown in figure 3.
The application for a Secure Logbook for Bluetooth Data Transfer on Mobile devices **is** an offline stand-alone application which can be described as portable integrated application device use in storing information sent via Bluetooth. Security and portability properties make the application a potential tool for secure transfer technique.

It is aimed at creating an efficient and secured working environment for storing or retrieving data sent via Bluetooth. To fulfil these requirements, this system should contain Bluetooth information that acquires processes and store data sent.

The client and server requires the Bluetooth discovery manager which is responsible for handling the connectivity, communication manager which deals with sending request to the server for searching content and receiving list of device containing required content and also the file resources which is used in reading file and the private folder locker which allows the user make content available to the other device. The server side for this research is the sending device, which is relational database system as shown the equation (1)

Figure 2: Architecture design for the system

The database used in the system is a relational database shown in equation (1)

$D = \{R_1 (S_1),... R_n (S_n)\}$

Such that

Where $R^n_{i=1}$ is the domain and S_1, $S_2 \ldots S_n$ is the attributes of the domain.

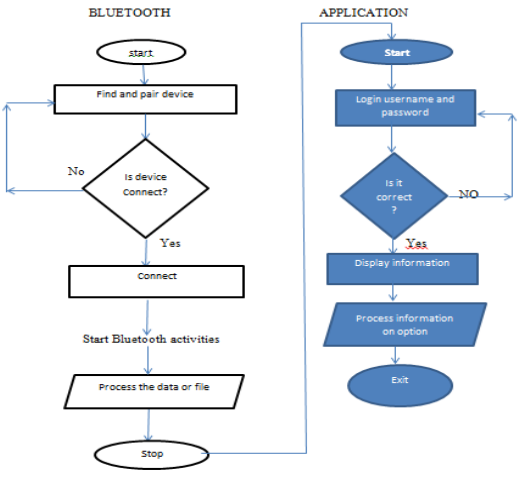

Figure 3: The flowchart diagram for the system

ii. SERVER SIDE

In order for client devices to find the server, it must be made discoverable by calling ensure Discoverable(). Once the server can be seen, then the quizzing service is started by calling setup

Service(). At this point, the server is started and sequentially selects a Universally Unique Identifier (UUID) from a queue of UUID's. A Bluetooth ServerSocket is created listening on the RFCOMM channel associated with this UUID. When a client opens a connection using the same UUID, a Bluetooth Socket is spawned for that client. The server then listens on another RFCOMM channel for another client. A blocking queue was used to store the UUIDS that are going to be used to get the RFCOMM channels. Since the accept() method is a system blocking call, it states that the device will not be able to perform anything else, hence only one server can be used. But the UUID's will be cycled to choose the correct RFCOMM.

iii. CLIENT SIDE

For the client application to work, the Bluetooth must be turned on. The client has to then searchfor the server by doing a device discovery. Scan For Devices () searches for all devices in close proximity to the client. A list of devices found is presented to the user. User interaction is required to select the server.

Once the server is selected, a service discovery is executed. This is where the client uses the same UUID as the server. The server then does a look in the service discovery database to ensure service is listed. Once the match is successful, the server sends the RFCOMM channel number on which the service is listening. The client then connects to the server using the COMM channel number. Data transfer to the server then begins.

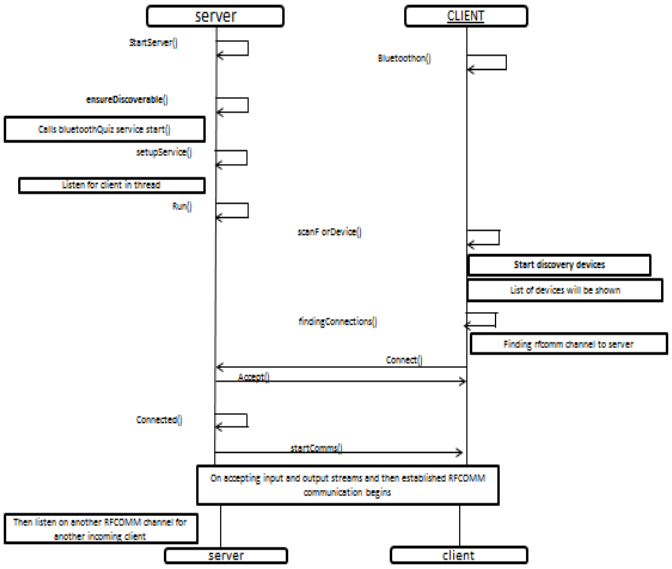

Figure 4: Sequence Diagram for Sever-Client device Bluetooth connection

iv. DATABASE

The database is the application (BLUElog), it retrieves and store information sent via Bluetooth. It displays the information logs (containing the address where the information is sent to, date and time). Security key is used to encrypt the information because of unauthorized access.

MATHEMATICAL MODEL

Let S_1, S_2, S_3,…, S_n be some attributed and R_1, R_2, R_3, …, R_n be their domain. A relation scheme relates certain attributes with their domain in context of their relation. A relation scheme can be represented as;

$$R = (S_1:R_1, S_2:R_2, S_3:R_3,\ldots,S_n:R_n) \quad (2)$$

Blogd scheme = (blid:Text, blname:Text, bladdress:Text, bltime:Text, bldate:Date) OR

BlogD (blid, blname, bladdress, bldate, bltime)

Whereas the blid, blname, bladdress, bldate, bltime are attributes and the text, text, text, text, text are their respective domains. A database as per this relation scheme can be:

BlogD = (blid:S001, blname:Jason, bladdress:30:AA:BD:D0:C6:CF, bldate:2016.06.11, bltime:12:35).

The above relation scheme if represented in dimensional structure called a table as shown in Table 1

Table 1

Blid	blname	bladdress	bldate	bltime
S001	Jason	30:AA:BD:D0:C6:CF	2016.06.11	12:35

DEFINITION OF TERMS

BLUElog: This is an application for keeping logs of file sent via Bluetooth.

RFCOMM: it provides the means of data communication to flow control the higher level layer. These layers can be end-user applications and higher level protocol.

CLIENT: a client is a computer hardware or software that accesses a service made available by a server.

SERVER: a server is both a running instance of some software that is capable of accepting request from client, and the computer that executes such software.

DATABASE: it is a collection of information that is organized so that it can be easily be accessed, managed, and updated.

IX. OPERATION OF THE SYSTEM

The system is to be carried by a Bluetooth connection, Bluetooth less energy is used in this operation because of it a low power consuming and has 2MHz channel. Android studio development tool used to implement the system.

The application is lunch on the device after been installed, the user open the application and created his/her password and recovery password. The information is shown to the user, the logs which include the address, time start and time end and the device which the information is sent to.

X. IMPLEMENTATION PROCEDURE

i. LOAD THE PROGRAM

At the outset, the initialization stage is the program setup, thereby pairing Bluetooth device together as shown in the figure 5 though to figure 10

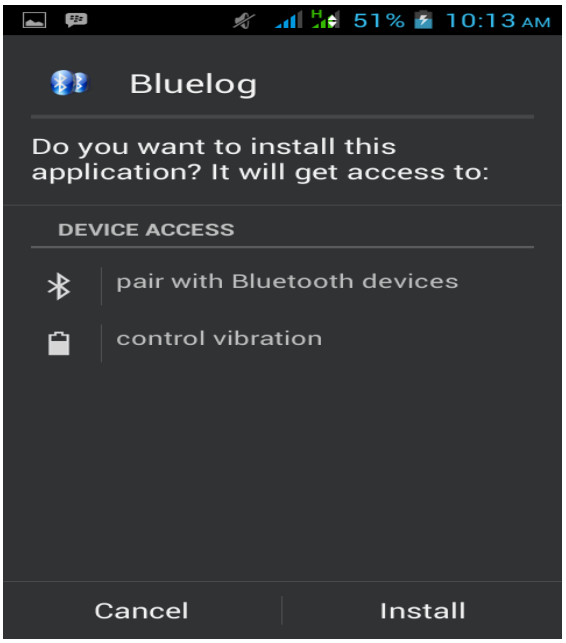

Figure 5 Initial setup and permission to install

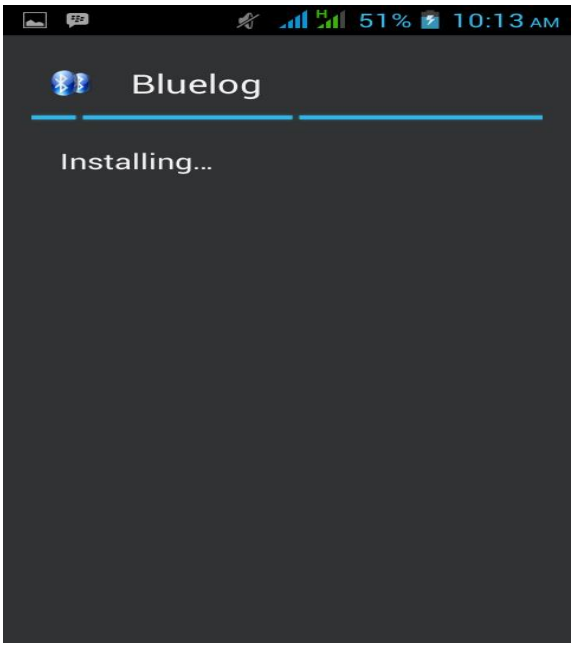

Figure 6 Displaying Installation progress

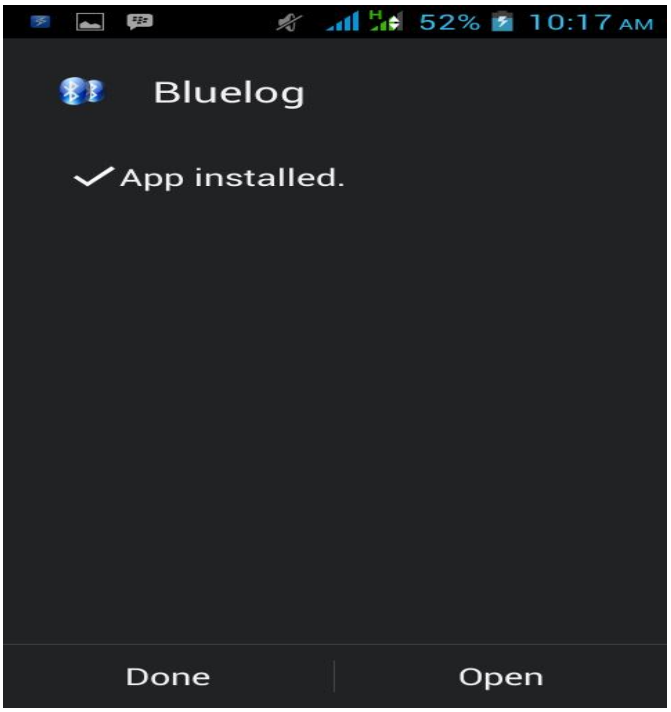

Figure 7 showing that application has been installed

ii. HOMEPAGE

After the welcome screen is lunched, the home page pops up with the request to create a login password and a recovery password for reset in the case the password is forgotten

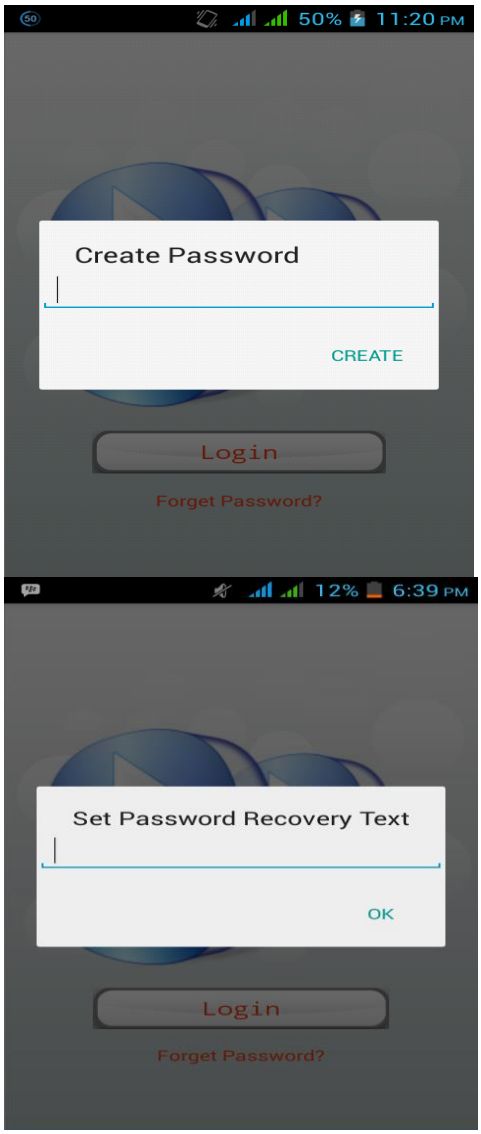

Figure 8: Screenshot of Homepage showing the create password

iii. THE LOGIN INTERFACE

The login screen interface is where the user supplies his/her password after creating one and access is granted if it authentic and denied it is not.

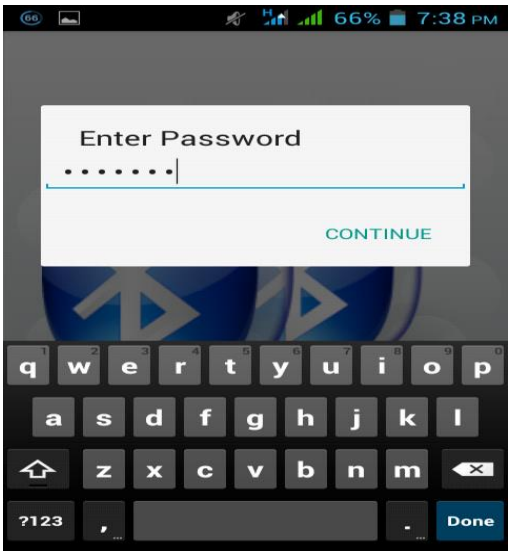

Figure 9: Screenshot of the login screen

iv. THE DISPLAY INTERFACE

This is the interface where information is displayed for user to process. Its shows the name of the device time start and time stop, date and address where the file is sent to.

Figure 10: Screenshot of Information Display

v. EXPERIMENT TEST

Bluelog testing was carried out by using the emulator in the Android developer kit. Bluelog application was installed and tested on three different Android mobile which are Techno, Samsung and Infinix mobile and it shows that the application is running perfectly and working efficiently.

XI. CONCLUSION

This research work presented a method to keep Bluetooth logs on android mobile device. It is a simple free application for keeping track of Data and files that are sent via Bluetooth connection on the mobile device. The application Bluelog is able to obtain and store data sent via Bluetooth.
So to say one of the limitations of this application is that when the details are stored, it does not identify what file is been sent maybe it a music, picture or document file.
For future enhancement to this application, improvement can be explored in the area of file identification. To identify what file is sent. Maybe it a music, video, picture or document file.

XII. ACKNOWLEDGEMENT

The contribution of Musediq Oluwafunke and Olajide Olamide Esther is acknowledged in the realization of this research.

XIII. REFERENCES

[1] Andrea Goldsmith. *Wireless Communications;* Cambridge University online publication 2012.

[2] Gunasekaran W., K.V.S.S.S.S Sairan, Rama Reddy; Bluetooth wireless communication. IEEE Communication Magazine 2002vol 40 issue 6. Page 90-96.

[3] Ramiro Jordan and Chaouki T. Abdallah (2012); Electrical & Computer Engineering. Department, The University of New Mexico, Albuquerque, New Mexico. page 185-193.

[4] www.litepoint.com/support (Bluetooth Low Energy) Retrieved 02-December 2015).

[5] Bluetooth SIG Website (2014). http//www.Bluetooth.org\ (Online; Retrieved 02-October-2015).

[6] Dhanashree Hasul and Deepak Nair. Design and Implementation of Mobile Bluetooth Hotspot; Journal of Emerging Technology and Advanced Engineering 2013). Volume 3, Issue 9. page 104-110.

[7] Saeed Ahmad. (2011); Developing a Bluetooth Ordering Application; Department of Applied Science. University of Vaasan Ammattikorkeakoulu, Finland. pages:1-47.

[8] Amrita Deb and Swarnabha Sinha; An Android Messenger app based on Bluetooth Connectivity; IOSR Journal of Computer Engineering 2014 Volume 16, Issue 3, Page 61-66.

[9] Claudio Palazzi, Armir Bujari and Emanuele Cervi.2007." P2P File Sharing on Mobile Phones; Department of pure and applied mathematics. University of padova, Italy. Page 1-5

[10] Gerard McNamara and Yanyan Yang (2008); Mobile P2P file sharing environment overBluetooth. Department of Electronic and Computer Engineering, University of Portsmouth, United Kingdom. page 863-868.